Name : KRISH SINGH

Roll: AID25072

Branch: AID Semester 2

1.In the following matrices, check whether the set of row vectors is independent or not. Are the set of column vectors independent?

```
% Qno 1 A
A=[1 2 3; 5 9 8; 5 5 -20];
% this method is valid for only n*n square matrices
if det(A) < 1e-10
    disp("A is dependent");
else
    disp("A is independent");
end
```

```
A is dependent
```

```
% Another method using rank
[M,N]=size(A)
```

```
M =
3
N =
3
```

```
if rank(A)==M
    disp("rows of A are Independent");
else
    disp("Rows of A are dependent");
end
```

```
Rows of A are dependent
```

```
if rank(A)==N
    disp("columns of A are Independent");
else
    disp("columns of A are dependent");
end
```

```
columns of A are dependent
```

```
%Qno 1 c
C=[1 2 1; 2 3 4; 5 1 2; 6 3 1];
[M,N]=size(C)
```

```
M =
4
N =
3
```

```
if rank(C)==M
```

```
    disp("Rows C are independent");
else
    disp("Rows C are dependent");
end
```

```
Rows C are dependent
```

```
if rank(C)==N
    disp("Columns of C are independent");
else
    disp("Columns of C are dependent");
end
```

```
Columns of C are independent
```

```
%Qno 1 b
B=[1 2 -1; -1 0 1; 2 1 -1];
[M,N]=size(B)
```

```
M =
3
N =
3
```

```
if rank(B)==M
    disp("Rows of B are independent");
else
    disp("rows of B are dependent");
end
```

```
Rows of B are independent
```

```
if rank(B)==N
    disp("Columns of B are independent");
else
    disp("Columns of B are dependent");
end
```

```
Columns of B are independent
```

```
%Qno 1 d
D=[1 2 -5; -1 0 1; 1 1 -1];
[M,N]=size(D)
```

```
M =
3
N =
3
```

```
if rank(B)==M
    disp("Rows of B are independent");
else
    disp("Rows of B are dependent");
```

```
end
```

```
Rows of B are independent
```

```
if rank(B)==N
    disp("Columns of B are independent");
else
    disp("Columns of B are dependent");
end
```

```
Columns of B are independent
```

```
%Qno 1 E
E=[1 1 2; 4 2 1; 2 2 4];
[M,N]=size(E)
```

```
M =
3
N =
3
```

```
if rank(E)==M
    disp("Rows of D are independent");
else
    disp("Rows of D are dependent");
end
```

```
Rows of D are dependent
```

```
if rank(E)==N
    disp("Columns of D are independent");
else
    disp("Columns of D are dependent");
end
```

```
Columns of D are dependent
```

2. Check whether the given set of matrices is linearly independent or not:

```
%Qno 2 a
A = [4 0; -2 -2];
B = [1 -1; 2 3];
C = [0 2; 1 4];
D = [6 -8; -1 -8];

M = [A(:), B(:), C(:), D(:)];
if rank(M)==size(M,2)
    disp("Lineraly independent")
else
    disp("Lineraly dependent")
```

```
end
```

```
Lineraly dependent
```

```
%Qno 2b
A = [1 1; 2 3];
B = [6 0; -1 4];
C = [3 0; 1 7];
D = [5 1; 4 2];
E = [7 1; 2 9];
M=[A(:),B(:),C(:),D(:),E(:)];
if rank(M)==size(M,2)
    disp("Linearly independent")
else
    disp("Linearly dependent")
end
```

```
Linearly dependent
```

```
%Qno 2 c
A = [3 6; 3 -6];
B = [0 -1; -1 0];
C = [0 -8; -12 -4];
D = [1 0; -1 2];
M=[A(:),B(:),C(:),D(:)];
if rank(M)==size(M,2)
    disp("Linearly Independent")
else
    disp("Linearly Dependent")
end
```

```
Linearly Independent
```

3. **Check whether the set of polynomials is linearly independent or not:**

```
%Qno 3a
P1=[0 1 0];
P2=[1 1 0];
P3=[1 1 -1];
M=[P1(:),P2(:),P3(:)];
if rank(M)==size(M,2)
    disp("Linearly independent");
else
    disp("Linearly dependent");
end
```

```
Linearly independent
```

```
%Qno 3b
p1=[1 1 1];
p2=[-1 -1 2];
p3=[3 2 -1];
M=[p1(:),p2(:),p3(:)];
if rank(M)==size(M,2)
    disp("Linearly Independent");
else
    disp("Linearly dependent");
end
```

```
Linearly Independent
```

```
%Qno 3c
p1=[1 1 1];
p2=[-1 -1 2];
p3=[3 2 -1];
p4=[1 1 -2];
M=[p1(:) p2(:) p3(:) p4(:)];
if rank(M)==size(M,2)
    disp("Linearly independent");
else
    disp("linearly dependent");
end
```

```
linearly dependent
```

4. Check whether the set of functions is linearly independent or not:

```
%Qno 4 a
syms x;
f1=sin(x);
f2=cos(x);
M=[f1 f2];
% construct Wronskian
W= det([M;diff(M,x)]);
%simplify Wronskian
W = simplify(W)
```

```
W = −1
```

```
if W~=0
    disp("functions are linearly Independent");
else
    disp("Functions are Linearly Dependent");
end
```

functions are linearly Independent

```
%Qno 4 b
f1=sin(x);
f2=cos(x);
f3=sin(1+x);
M=[f1 f2 f3];
W=det([M;diff(M,x);diff(M,x,2)]);
W=simplify(W)
```

W = 0

```
if W~=0
    disp('functions are Linearly Independent');
else
    disp('Functions are Linearly Dependent');
end
```

Functions are Linearly Dependent

```
%Qno 4 c
% Qno 4(c)

syms x
f1 = sin(x);
f2 = cos(x);
f3 = tan(x);
M = [f1 f2 f3];
W = simplify(det([M; diff(M,x); diff(M,x,2)]));
if W~=0
    disp('Functions are Linearly Independent');
else
    disp('Functions are Linearly Dependent');
end
```

Functions are Linearly Independent

```
%Qno 4 d
f1=sin(x);
f2=cos(x);
f3=x;
M=[f1 f2 f3];
W=det([M;diff(M,x);diff(M,x,2)]);
W=simplify(W)
```

W = $-x$

```
if W~=0
    disp('Functions are Linearly Independent');
```

```
else
    disp('Functions are Linearly Dependent');
end
```

```
Functions are Linearly Independent
```

5. Check whether the set of objects given in questions 1, 2, 3 and 4 forms a basis or not.

```
%Qno 5 for 1A
A = [1 2 3; 5 9 8; 5 5 -20];

r = rank(A);
[n, m] = size(A);
if r == m
    disp('Column vectors form a BASIS for column space');
else
    disp('Column vectors do NOT form a basis');
end
```

```
Column vectors do NOT form a basis
```

```
if r == n
    disp('Row vectors form a BASIS for row space');
else
    disp('Row vectors do NOT form a basis');
end
```

```
Row vectors do NOT form a basis
```

```
%Qno 5 for 1B
B=[1 2 -1; -1 0 1; 2 1 -1];
r = rank(B);
[n, m] = size(B);
if r == m
    disp('Column vectors form a BASIS for column space');
else
    disp('Column vectors do NOT form a basis');
end
```

```
Column vectors form a BASIS for column space
```

```
if r == n
    disp('Row vectors form a BASIS for row space');
else
    disp('Row vectors do NOT form a basis');
end
```

```
Row vectors form a BASIS for row space
```

```
%Qno 5 for 1C
C=[1 2 1; 2 3 4; 5 1 2; 6 3 1];
r = rank(C);
[n, m] = size(C);
if r == m
    disp('Column vectors form a BASIS for column space');
else
    disp('Column vectors do NOT form a basis');
end
```

```
Column vectors form a BASIS for column space
```

```


if r == n
    disp('Row vectors form a BASIS for row space');
else
    disp('Row vectors do NOT form a basis');
end
```

```
Row vectors do NOT form a basis
```

```
%Qno 5 for 1D
D=[1 2 -5; -1 0 1; 1 1 -1];
r = rank(D);
[n, m] = size(D);
if r == m
    disp('Column vectors form a BASIS for column space');
else
    disp('Column vectors do NOT form a basis');
end
```

```
Column vectors form a BASIS for column space
```

```


if r == n
    disp('Row vectors form a BASIS for row space');
else
    disp('Row vectors do NOT form a basis');
end
```

```
Row vectors form a BASIS for row space
```

```
%Qno 5 for 1E
E=[1 1 2; 4 2 1; 2 2 4];
r = rank(E);
[n, m] = size(E);
if r == m
```

```
    disp('Column vectors form a BASIS for column space');
else
    disp('Column vectors do NOT form a basis');
end
```

```
Column vectors do NOT form a basis
```

```


if r == n
    disp('Row vectors form a BASIS for row space');
else
    disp('Row vectors do NOT form a basis');
end
```

```
Row vectors do NOT form a basis
```

```
%Qno 5 for 2A
A = [4 0; -2 -2];
B = [1 -1; 2 3];
C = [0 2; 4 3];
D = [6 -8; -1 -8];

M = [A(:) B(:) C(:) D(:)];

if rank(M) == size(M,2)
    disp('Matrices form a BASIS');
else
    disp('Matrices do NOT form a basis');
end
```

```
Matrices form a BASIS
```

```
%Qno 5 for 2B
A = [1 1; 2 3];
B = [6 0; -1 4];
C = [3 0; 1 7];
D = [5 1; 4 2];
E = [7 1; 2 9];
M=[A(:),B(:),C(:),D(:),E(:)];
if rank(M) == size(M,2)
    disp('Matrices form a BASIS');
else
    disp('Matrices do NOT form a basis');
end
```

```
Matrices do NOT form a basis
```

```
%Qno 5 for 2C
A = [3 6; 3 -6];
B = [0 -1; -1 0];
```

```
C = [0 -8; -12 -4];
D = [1 0; -1 2];
M=[A(:),B(:),C(:),D(:)];
if rank(M) == size(M,2)
    disp('Matrices form a BASIS');
else
    disp('Matrices do NOT form a basis');
end
```

```
Matrices form a BASIS
```

```
%Qno 5 for 3A
P1=[0 1 0];
P2=[1 1 0];
P3=[1 1 -1];
M=[P1(:),P2(:),P3(:)];
if rank(M) == size(M,2)
    disp('Polynomials form a BASIS');
else
    disp('Polynomials do NOT form a basis');
end
```

```
Polynomials form a BASIS
```

```
%Qno 5 for 3B
p1=[1 1 1];
p2=[-1 -1 2];
p3=[3 2 -1];
M=[p1(:),p2(:),p3(:)];
if rank(M) == size(M,2)
    disp('Polynomials form a BASIS');
else
    disp('Polynomials do NOT form a basis');
end
```

```
Polynomials form a BASIS
```

```
%Qno 5 for 3C
p1=[1 1 1];
p2=[-1 -1 2];
p3=[3 2 -1];
p4=[1 1 -2];
M=[p1(:) p2(:) p3(:) p4(:)];
if rank(M)==3 && size(M,2)==3
    disp('Polynomials form a BASIS for P2');
else
    disp('Polynomials do NOT form a basis for P2');
end
```

Polynomials do NOT form a basis for P2

```
%Qno 5 for 4A
syms x
f1=sin(x);
f2=cos(x);
M=[f1 f2];
% construct Wronskian
W= det([M;diff(M,x)]);
%simplify Wronskian
W = simplify(W)
```

W = -1

```
if W==0
    disp("Does not form Basis");
else
    disp("Does form Basis");
end
```

Does form Basis

```
%Qno 5 for 4B
f1=sin(x);
f2=cos(x);
f3=sin(1+x);
M=[f1 f2 f3];
W=det([M;diff(M,x);diff(M,x,2)]);
W=simplify(W)
```

W = 0

```
if W==0
    disp("Does not form Basis");
else
    disp("Does form Basis");
end
```

Does not form Basis

```
%Qno 5 for 4C
f1 = sin(x);
f2 = cos(x);
f3 = tan(x);

W = det([
    f1 f2 f3;
    diff(f1,x) diff(f2,x) diff(f3,x);
    diff(f1,x,2) diff(f2,x,2) diff(f3,x,2)
```

```
]);

simplify(W)
```

```
ans =
```

$$\frac{\sin(x)\ (\sin(x)^2-3)}{\cos(x)^3}$$

```
if W==0
    disp("Does not form Basis");
else
    disp("Does form Basis");
end
```

```
Does form Basis
```

```
%Qno 5 for 4D
f1=sin(x);
f2=cos(x);
f3=x;
M=[f1 f2 f3];
W=det([M;diff(M,x);diff(M,x,2)]);
W=simplify(W)
```

```
W =
```

$$-x$$

```
if W==0
    disp("Does not form Basis");
else
    disp("Does form Basis");
end
```

```
Does form Basis
```

6. Generate a 5x5 matrix A of rank 4.

```
A=randi([0,9],5,4)*randi([0,9],4,5)
```

```
A = 5×5
    22    21    32    32    37
    82    80    97   119   123
    89    81    99   110   104
    47    37    79    44    52
    86    75   125    96   102
```

```
rank(A)
```

```
ans =
4
```

a. Retrieve an element with row index 3, and column index 5.

```
A(3,5)
```

```
ans =
104
```

b. Retrieve first row from A and store in b

```
b=A(1,:)
```

```
b = 1×5
    22    21    32    32    37
```

c. Retrieve first and third row from A and store in C.

```
C=A([1,3],:)
```

```
C = 2×5
    22    21    32    32    37
    89    81    99   110   104
```

d. Retrieve second column from A and store in b

```
b=A(:,2)
```

```
b = 5×1
    21
    80
    81
    37
    75
```

e. Retrieve second and fourth column from A and store in C.

```
C=A(:,[2,4])
```

```
C = 5×2
    21    32
    80   119
    81   110
    37    44
    75    96
```

7. **Find the basis for the null space, row space and column space of the matrices in Q1**

```
%Qno 7 for 1A
A=[1 2 3; 5 9 8; 5 5 -20];
[R, pivot_cols] = rref(A);
ColBasis = A(:, pivot_cols)
```

```
ColBasis = 3×2
```

```
     1     2
     5     9
     5     5
```

```
RowBasis = R(any(R,2), :)
```

```
RowBasis = 2×3
     1     0   -11
     0     1     7
```

```
NullBasis = null(A, 'r')
```

```
NullBasis = 3×1
    11
    -7
     1
```

```
%Qno 7 for 1B
B=[1 2 -1; -1 0 1; 2 1 -1];
[R, pivot_cols] = rref(B);
ColBasis = B(:, pivot_cols)
```

```
ColBasis = 3×3
     1     2    -1
    -1     0     1
     2     1    -1
```

```
RowBasis = R(any(R,2), :)
```

```
RowBasis = 3×3
     1     0     0
     0     1     0
     0     0     1
```

```
NullBasis = null(B, 'r')
```

```
NullBasis =

  3×0 empty double matrix
```

```
%Qno 7 for 1C
C=[1 2 1; 2 3 4; 5 1 2; 6 3 1];
[R, pivot_cols] = rref(C);
ColBasis = C(:, pivot_cols)
```

```
ColBasis = 4×3
     1     2     1
     2     3     4
     5     1     2
     6     3     1
```

```
RowBasis = R(any(R,2), :)
```

```
RowBasis = 3×3
     1     0     0
     0     1     0
     0     0     1
```

```
NullBasis = null(C, 'r')
```

```
NullBasis =

  3×0 empty double matrix
```

```
%Qno 7 for 1D
D=[1 2 -5; -1 0 1; 1 1 -1];
[R, pivot_cols] = rref(D);
ColBasis = D(:, pivot_cols)
```

```
ColBasis = 3×3
     1     2    -5
    -1     0     1
     1     1    -1
```

```
RowBasis = R(any(R,2), :)
```

```
RowBasis = 3×3
     1     0     0
     0     1     0
     0     0     1
```

```
NullBasis = null(D, 'r')
```

```
NullBasis =

  3×0 empty double matrix
```

```
%Qno 7 for 1E
E=[1 1 2; 4 2 1; 2 2 4];
[R, pivot_cols] = rref(E);
ColBasis = E(:, pivot_cols)
```

```
ColBasis = 3×2
     1     1
     4     2
     2     2
```

```
RowBasis = R(any(R,2), :)
```

```
RowBasis = 2×3
    1.0000         0   -1.5000
         0    1.0000    3.5000
```

```
NullBasis = null(E, 'r')
```

```
NullBasis = 3×1
    1.5000
   -3.5000
    1.0000
```

8. To check if a given vector **y** is in any of the subspaces generated by a matrix A.

```
%Qno 8 a i
```

```
A = [1 3 4 7;2 4 6 10; 3 5 8 13; 4 6 10 16];
u = [-2;-3;1;1];

if rank([A u]) == rank(A)
    disp('u is in the column space of A');
end
if A*u == zeros(size(A,1),1)
    disp('u is in the null space of A');
end
```

```
u is in the null space of A
```

```
if A'*u == zeros(size(A,2),1)
    disp('u is in the left null space of A');
end
if rank([A' u]) == rank(A')
    disp('u belongs to the ROW SPACE of A');
end
```

```
%Qno 8 a ii
v=[5;8;11;14];

if rank([A v]) == rank(A)
    disp('v is in the column space of A');
end
```

```
v is in the column space of A
```

```
if A*v == zeros(size(A,1),1)
    disp('v is in the null space of A');
end
if A'*v == zeros(size(A,2),1)
    disp('v is in the left null space of A');
end
if rank([A' v]) == rank(A')
    disp('v belongs to the ROW SPACE of A');
end
```

```
%Qno 8 a iii
w=[1;1;2;3];

if rank([A w]) == rank(A)
    disp('w is in the column space of A');
end
if A*w == zeros(size(A,1),1)
    disp('w is in the null space of A');
end
if A'*w == zeros(size(A,2),1)
    disp('w is in the left null space of A');
end
```

```
if rank([A' w]) == rank(A')
    disp('w belongs to the ROW SPACE of A');
end
```

```
w belongs to the ROW SPACE of A
```

```
%Qno 8 a iv
y=[1;2;0;-1];

if rank([A y]) == rank(A)
    disp('y is in the column space of A');
end
if A*y == zeros(size(A,1),1)
    disp('y is in the null space of A');
end
```

```
y is in the null space of A
```

```
if A'*y == zeros(size(A,2),1)
    disp('y is in the left null space of A');
end
if rank([A' y]) == rank(A')
    disp('y belongs to the ROW SPACE of A');
end
```

```
%Qno 8 a v
m=[-1;1;1;-1];

if rank([A m]) == rank(A)
    disp('m is in the column space of A');
end
if A*m == zeros(size(A,1),1)
    disp('m is in the null space of A');
end
if A'*m == zeros(size(A,2),1)
    disp('m is in the left null space of A');
end
```

```
m is in the left null space of A
```

```
if rank([A' m]) == rank(A')
    disp('m belongs to the ROW SPACE of A');
end
```

```
%Qno 8 b i
A=[1 -1 2 3;0 2 1 4;1 1 3 1;2 0 5 4];
v=[5;1;-2;0];

if rank([A v]) == rank(A)
    disp('v is in the column space of A');
```

```
end
if A*v == zeros(size(A,1),1)
    disp('v is in the null space of A');
end
```

```
v is in the null space of A
```

```
if A'*v == zeros(size(A,2),1)
    disp('v is in the left null space of A');
end
if rank([A' v]) == rank(A')
    disp('v belongs to the ROW SPACE of A');
end
```

```
%Qno 8 b ii
w=[0;2;2;2];

if rank([A w]) == rank(A)
    disp('w is in the column space of A');
end
```

```
w is in the column space of A
```

```
if A*w == zeros(size(A,1),1)
    disp('w is in the null space of A');
end
if A'*w == zeros(size(A,2),1)
    disp('w is in the left null space of A');
end
if rank([A' w]) == rank(A')
    disp('w belongs to the ROW SPACE of A');
end
```

```
%Qno 8 b iii
u=[-1;2;-1;1];

if rank([A u]) == rank(A)
    disp('u is in the column space of A');
end
if A*u == zeros(size(A,1),1)
    disp('u is in the null space of A');
end
if A'*u == zeros(size(A,2),1)
    disp('u is in the left null space of A');
end
if rank([A' u]) == rank(A')
    disp('u belongs to the ROW SPACE of A');
end
```

```
%Qno 8 b iv
m=[3;-1;7;7];

if rank([A m]) == rank(A)
    disp('m is in the column space of A');
end
if A*m == zeros(size(A,1),1)
    disp('m is in the null space of A');
end
if A'*m == zeros(size(A,2),1)
    disp('m is in the left null space of A');
end
if rank([A' m]) == rank(A')
    disp('m belongs to the ROW SPACE of A');
end
```

```
m belongs to the ROW SPACE of A
```

9. Using rref(A) command , how will you retrieve the set of independent columns (columns with pivotal element produced out of row elimination process) in C and nonzero-rows produced out of row elimination process in R. Note that the number of pivotal columns in A and number of non-zero rows will be same.

1. Answer :
2. [RR, ic]=rref(A);
3. r = length(ic);
4. R=RR(1:r, :);
5. C=A(:, ic);

% It can be observed that A=C*R

```
% Question No 9
A = [1 2 3;2 4 6;1 1 1];
[RR, ic] = rref(A);
r = length(ic);
R = RR(1:r, :)
```

```
R = 2×3
     1     0    -1
     0     1     2
```

```
C = A(:, ic)
```

```
C = 3×2
     1     2
     2     4
     1     1
```

```
% Verification
C*R
```

```
ans = 3×3
     1     2     3
     2     4     6
     1     1     1
```